Available online at www.sciencedirect.com
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 00 (2011) 000–000
Procedia Engineering 15 (2011) 1642 – 1646
Procedia Engineering
Procedia Engineering
www.elsevier.com/locate/procedia
www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

The Clustering Algorithm of Query Result based on Maximal Frequent

WEI Yu-wei *

[2] *Faculty Electromechanical Engineering,Guangdong University of Technology,Guangzhou 510006, China*

Abstract

Most of existing web page clustering algorithms is based on short and uneven snippets of web page, which often cause bad clustering performance. On the other hand, the classical clustering algorithm for full text web pages is too complex to provide good cluster label in addition to the incapability on-line clustering. To address above problems, this article presents an on-line web page clustering algorithm based on maximal frequent item sets (MFIC). At first, the maximal frequent item sets are mined, and then the web pages are clustered based on shared frequent item sets. Secondly, clusters are labelled based on the frequent items. Experimental results show that MFIC can effectively reduce clustering time, improve clustering accuracy by 15%, and generate understandable labels.

"Keywords: Search Engine;Frequent Itemsets;Page Clustering"

1. Introduction

With the increasing of internet information, SE (Search Engine) becomes the indispensable tools. Now the most general SE sort the WebPages based on the correlation degree to the user enquiry, and return the results to user with a list view. The users ought to judge every webpage whether the results are satisfied with their demand. The research shows the most users make use of the short and uncertain search string. But 85% users only view the results of the first page, and 78% user never change their research terms. In

* Corresponding author. Tel.:+8613602446699
E-mail address: weiyuwei@gdut.edu.cn

doi:10.1016/j.proeng.2011.08.306

addition, because of their different background, the results desired are different. Therefore, in order to meet the requirements of user's query quality increasingly, the user wants to improve the usability of query results.

In order to solve the problem, this article puts forward a online clustering algorithms of query results based on webpage maximal frequent itemsets. By improving the mining algorithms of maximal frequent itemsets, it can be use for the online clustering of he SE query results. New algorithms uses the sharing relation of webpage sets and frequent itemsets to cluster, meanwhile describes the clear and definite tags of every category. The experiment results show that the clustering based on maximal frequent itemsets can reduce the clustering time based on full-text, at the same time the clustering accuracy can improve 15% or so.

2. Maximal Frequent Itemset Mining

2.1. The Basic Concept of Frequent Itemsets

Files should be in MS Word format only and should be formatted for direct printing. Figures and tables should be embedded and not supplied separately. Please make sure that you use as much as possible normal fonts in your documents. Special fonts, such as fonts used in the Far East (Japanese, Chinese, Korean, etc.) may cause problems during processing.

In order to adopt maximal frequent itemset as the basic feature of on-line cluster algorithm based on full-text webpage. In this section, it firstly introduces the basic concept of frequent itemset,, a detailed introduction related frequent itemset refers to other document literature.

Definition 1: Assuming $I=\{I_1,I_2,\ldots I_n\}$ is a set of n different items. For a set X, $X\subseteq I$ and $k=|X|$, the X is called as k itemset, the length of X is the amount of including items, it is k.

Definition 2: $D=\{T_1,T_2,\ldots,T_m\}$ is a set of m different transactions, among $T_i\subseteq I$. For the given the transaction set D, define the support of X is the amount of transaction occurred X, named as $Sup(X)$. User may define a minimal support counting, $\min\_supp$, it is either absolute counting or relative counting.

Definition 3: Supposing the transaction set D and the minimal support counting $\min\_supp$, for itemsets, $X\subseteq I$, if $Sup(X)>\min\_supp$ and $\forall(Y\subseteq I\wedge X\subseteq Y)$, $Sup(Y)<\min\_supp$, then X is named as the maximal frequent itemset in the transaction set D.

In this article, the transaction set is the webpage sets of query results, every webpage is a transaction. Itemset is the sets included terms in the webpage, the terms of webpage is the item of transaction.

2.2. Maximal Frequent Itemsets Mining Algorithm

The common algorithm of frequent itemset mining is FP-Growth algorithm. It firstly constructs a FP-Tree that is a threaded tree structure to storage the transaction of sets.[3] The construction of FP-Tree firstly makes a statistic for support counting of all items, these items its support exceeded minimal support counting arrange in Header table of FP-Tree in decreasing order. Then every time only read in a transaction, and map to FP-Tree routing. Fig. 1 is a example of FP-Tree (its support is 2), among (a) expresses the transaction sets, (b) is the FP-Tree constructed. In this figure, solid line presents the routing of transaction mapping tree, the dotted line points to the location in the tree from Header Table, the counting of node expresses the support corresponding to itemset in the ending routine from root node to current node, such as the node "trademark: 2" presents the support of this itemset {car, Geely, trademark} is 2.

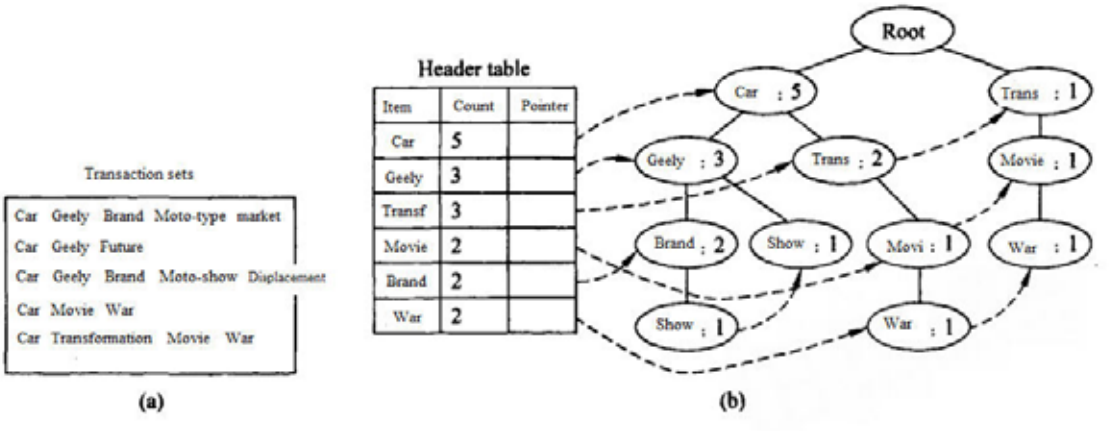

Fig.1. a example of FP-Tree (min_supp = 2)

3. Query Result Cluster Algorithm Based on Maximal Frequent Itemsets

After mining the frequent itemsets, there are two ways to cluster: First, adopt the alternative word of frequent itemset to create the feature vector of webpage and use the traditional clustering algorithm based on vector space model. Second, cluster with the relation of frequent itemset overlap webpage set.[4] The former has been proved that the time complexity cannot satisfy the demand of on-line cluster, at the same time, the cluster e

The clustering algorithm introduced by this article adopts the relation of the webpage sharing the maximal frequent itemset to cluster. For the purpose of this article following research, some definitions as follow: $D=\{T_1,T_2,\ldots,T_m\}$ is the set of all transactions, and it is the set of query webpage in this article. $I=\{I_1,I_2,\ldots,I_n\}$ is the set of all item, and it is the set of terms included in webpage sets. $S_M=\{M_1,M_2,\ldots,M_n\}$ is the set of all maximal frequent itemsets mined in webpages, the webpage overlapped by a maximal frequent itemset M_i names as P_i, $P_i \subseteq D$. The process of clustering means that the set of webpages are divided into some clusters, named as $C=\{C_1,C_2,\ldots,C_l\}$, it is the set of cluster. The webpage sets included by a cluster C_i marks CP_i, $CP_i \subseteq D$, the set of maximal frequent itemsets included names as CM_i, $CM_i \subseteq S_m$, the set of frequent itemsets included is CI_i, $CI_i \subseteq I$. $D_c=\{T_1,T_2,\ldots,T_k\}$ is the webpage set overlapped by cluster. Below introduces the core steps of cluster algorithm.

Step 1: The generation of cluster

The longer the length of frequent itemset, the more the terms included, and the better expresses a detail topic, so the long cluster generated by frequent itemset is given priority to select.

The frequent itemset among S_m sorts in the order of their length, and in proper sequence select the longest frequent itemset M_i to generate cluster C_i, CP_i is the webpage set included by C_i, and it is the webpage set P_i overlapped by M_i, record the webpage set overlapped by cluster, $D_c = D_c \cup P_i$. In order to improve the speed of cluster generation, reduce the transmission effects in subsequent merging procedure, and further filter the frequent itemset of S_m. If a frequent itemset M_k overlaps the webpage set $P_k \subseteq D_c$, it means that all webpages of P_k have been overlapped by clusters, and doesn't generate the cluster C_k correspond to M_k.

Step 2: The merging of clusters

The clusters originally generated are more, and there are a lot of overlapping, so need to merge and generate the final cluster. The merging of clusters means that the clusters with high similarity merge a

cluster; usually the similarity of clusters is judged with the similarity of webpage sets included. For the clustering algorithm based on frequent itemset, the frequent itemset included by cluster is the important feature of cluster. The similarity of cluster is computed with the similarity of request itemset included.[5] In order to improve the accuracy, adopt the formula (1) to compute the similarity of clusters.

The similarity of cluster C_i and C_j names as $Sim(C_i, C_j)$, the similarity of webpage included names as $SimP_{ij}$, the similarity of frequent itemset included names as $SimI_{ij}$.

$$Sim(C_i, C_j) = \frac{|CP_i \cap CP_j|}{\min(|CP_i|, |CP_j|)} + \frac{|CI_i \cap CI_j|}{\min(|CI_i|, |CI_j|)} \quad (1)$$

The more $Sim(C_i, C_j)$, the higher the cluster C_i and C_j, and intends to merge.

Step 3: The cluster purification

The clusters are subdivided into the hard clusters and the soft clusters. The former demands a webpage only is belong to a category, the latter allows a webpage to belong to multiple category. So the hard clusters can reflect reality well. Because of the transmission effects of clusters merging, the clusters sometimes include some non-correlated webpages. It is a vital problem how to recognize the webpage of clusters is non-correlated webpage or multiple category webpage. In this article, the recogniton of non-correlated webpage is judged by the support webpage relative to cluster. So this article defines the support as fellow, the webpage P relative to cluster C_i.

$$Supp(P, C_i) = \sum_{M \in CM_i} |M| \times f(P, M) \quad (3)$$

According the the experiment, a empirical value can be set. When $Supp(P, C_i)$ is less than this value, it is the non-correlative webpage, and it would be delete from the clusters.

4. Experimental Results and Analysis

4.1. Experimental Condition and Experimental Data

The experimental data is the data set responding to 8 ambiguous query terms. For SE, and gain the union set. Then mark the participle and the word characteristic to the webpages, construct the index and keep the participle results for latter algorithm. Above-mentioned work is off-line completed, and prepares the data for on-line query clusters. The webpage set is manual marked category, every query term webpage set marks several categories.

T Because the K-Means algorithm demands to set k value, separately set 4 k value (5,6,7,8) to experiment, and every query get the highest F value as the final result in 4 experimental result. STC, Lingo and MFIC can automatic generate an arbitrary number of categories, and there are some clusters only including 2~3 webpages, but in practical application usually only shows the cluster including major webpages, the category less than 5 webpages names as other category in actual experiment. According to changing the parameter, the category number of 3 algorithms ranges from5 to 10.

4.2. Experimental Condition and Experimental Data

In this article, the experiment compares with MFIC based on full text and K-Means, at the same time compares the cluster-time with STC based on abstract.[6] For the full text of webpage, the cluster-time is too long to apply for on-line cluster. The experimental data displays that the time is more than 10 seconds.

In addition, the Lingo adopts the open java experiment, and other algorithms implement with C++, so they are compared with it.

The graph suggests the MFIC cluster-time is superior to K-Means. Because the MFIC cluster is based on webpage full text, it is a foregone conclusion that the cluster-time is longer than STC based on abstract. The experimental result shows that the MFIC can satify with the demand of on-line cluster if its cluster-time is about 2 seconds. In order to improve on system response, they can set the cache of cluster result and reduce the user waiting time in detail application

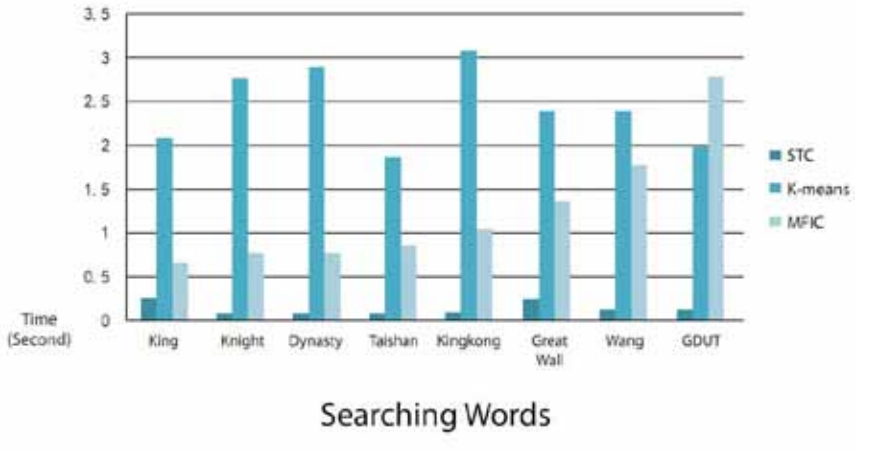

Fig.3. The cluster algorithm time comparison

Conclusion

This article proposes a cluster algorithm of SE returned results based on full text maximal frequent itemsets. Firstly, research the mining algorithm of frequent itemsets, and improve on the maximal frequent itemset mining combining FPMax algorithm, at the same time increase the mining speed. Then it proposes a MFIC algorithm based on maximal frequent itemset. The MFIC algorithm mostly includes three steps. Firstly, generate the cluster with the mined maximal frequent itemset. Secondly, merge and judge the clusters combining the similarity of frequent itemset with the similarity of document sets included by cluster. Finally, propose a label generation algorithm combining frequent itemset with terms sequence.

References

[1] Liping Jing, Michael K. Ng. An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(8), p.1026-1040.

[2] Wei Song, Soon Cheol Park. Genetic Algorithm based text clustering technique. Automatic evolution of cluster with high efficiency. 7th International Conference on Web-Age Information Management Workshops. Hong Kong, 2006, p.17-18.

[3] Daniel Crabtree, Xiaoying Gao. Improving Web Clustering by Cluster Selection. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence. 2005, p.172-178.

[4] Hung Chim, Xiaotie Deng. A New Suffix Tree Similarity Measure for Document Clustering. World Wide Web Conference Committee. 2007, p.121-129

[5] Daniel Crabtree, Peter Andreae. Query Directed Web Page Clustering. Proceeding of the IEEE/WIC/ACM International Conference on Web Intelligence. 2006, p.202-210.